SAM-TR-65-59

FITTING MULTI-COMPONENT EXPONENTIAL DECAY CURVES BY DIGITAL COMPUTER

EARL L. BELL, M.S.
RAUL GARCIA, B.A.

August 1965

USAF School of Aerospace Medicine
Aerospace Medical Division (AFSC)
Brooks Air Force Base, Texas

FITTING MULTI-COMPONENT EXPONENTIAL DECAY CURVES BY DIGITAL COMPUTER

EARL L. BELL, M.S.
RAUL GARCIA, B.A.

FOREWORD

This report was prepared in the Biometrics Branch under task No. 631902. The paper was submitted for publication on 11 June 1965.

The many useful suggestions and encouragement extended by Dr. M. Bryan Danford and Dr. Robert G. Rossing are gratefully acknowledged.

This report has been reviewed and is approved.

Harold V. Ellingson

HAROLD V. ELLINGSON
Colonel, USAF, MC
Commander

ABSTRACT

The mechanical-graphical "peel-off" method and Marquardt's composite Gauss-Newton and gradient iterative method were programmed for the Philco 2000, a 16K asynchronous digital computer. Both programs were coded in the Philco Algebraic Programming Language (ALTAC) using single-precision floating-point arithmetic.

Background material, flow charts, flow chart descriptions, subprogram usage, computer memory requirements, and illustrative numeric examples of the analyses of both simulated and empirical data are given. Each sample of simulated data possessed an error component; the effects of an asymptote, in several instances, were included during the generation of the data. Dog lung nitrogen washout activity experiments were the source of the empirical data.

FITTING MULTI-COMPONENT EXPONENTIAL DECAY CURVES BY DIGITAL COMPUTER

I. INTRODUCTION

The principal purpose of this paper is to discuss the automation of two nonlinear parameter estimation procedures—the classical "peel-off" and Marquardt's algorithm (2). As a mechanical-graphical method, the "peel-off" procedure yields good results for parameter estimation in a hypothesized mathematical model of a linear combination of exponential functions. But the length of time required in the application of this method is, in general, too long. Automation of the "peel-off' method rectified the time problem, but the computer-produced parameter estimates turn out to be somewhat inferior in accuracy. A parameter-estimate-refining program was written to improve the "peel-off" estimates. The refining process was accomplished through the adaptation of an algorithm that was described by Marquardt (2) for obtaining least-squares estimates of nonlinear parameters. Two iterative methods, classical Gauss-Newton and gradient, were combined. This combination yielded an iterative method with strong convergence properties and a compromise between two levels of rapidity of convergence. Both curve-fitting programs were coded in the Philco Algebraic Programming Language (ALTAC) (3), using single-precision floating-point arithmetic. The computational mode has a range from slightly more than 10^{+600} to slightly less than 10^{-60} and an accuracy of ten significant digits. The programs were tested on simulated as well as on dog lung nitrogen washout data.

Four sections follow this introductory section. The mathematical model used in fitting a linear combination of exponentials is briefly discussed in section II. The reader is referred to Danford (1) for a general discussion of exponential model equations. Background material, flow chart, flow chart description, subprogram usage, computer memory requirements, and numerical examples of the analyses of simulated data (generated with and without the effects of an asymptote and with error) and dog lung nitrogen washout data pertinent to the "peel-off method are covered in section III. Material similar to that of section III, on the composite Gauss-Newton and gradient method, is to be found in section IV. Section V is concerned with comments on such selected material as: adapting the computer programs to meet the user's requirements; possible places for improvement in the "peel-off" method program; the consequence of failing to use the best possible estimate of the asymptote; the use of smoothed data to improve parameter estimation; the importance of refining the preliminary parameter estimates produced by the "peel-off" method program; and the area in which the use of double-precision floating-point arithmetic may become necessary in the Gauss-Newton and gradient iterative process.

II. MATHEMATICAL MODEL

Our main results concern the estimation of the parameters in the model

$$\begin{aligned} Y(x) &= y(x) + \epsilon(x) \\ &= \alpha_0 + \sum_{m=1}^{N} \alpha_m \exp(-\beta_m x) + \epsilon(x) \\ &= \alpha_0 + \sum_{m=1}^{N} \alpha_m \omega_m{}^x + \epsilon(x), \end{aligned} \tag{1}$$

using the data points (x_i, Y_i), $i = 1,2,\ldots,L$, where

$y(x)$: True value of Y at x,
α_0 : Constant term or asymptote,
α_m, β_m : Model parameters > 0, $m = 1,2,\ldots,N$,
$\epsilon(x)$: Error term: NID $(0,\sigma)$ for each x and
σ : $\rho\, y(x)$,
ρ : Positive number; 100ρ can be viewed as a percent error,
L : Total number of data points,
N : Number of exponential components.

III. "PEEL-OFF" METHOD

Mechanical-graphical version

The "peel-off" method for parameter estimation has been used for a considerable period of time without undergoing any major modification. Its users during this time have probably numbered in the hundreds and this extensive usage has no doubt been due to the absence of a superior technic. Application of the method is quite easy. Simple tools such as a pencil, semilogarithmically scaled paper, and five-place common logarithm tables are needed plus "good" judgment on the part of the analyst. The method can become very tedious when many parameter estimates per experiment are required or there are many experiments to be analyzed. One must not fail to mention disadvantages of a graver nature: personal bias may be present that will affect the fit; the use of imperfect semilogarithmic paper may contribute to the bias in the estimates; the errors of estimation may be cumulative in nature; and finally there is the problem of securing an estimate of error.

A simplified summary of the sequence of steps taken by the data analyst in applying the "peel-off" method is as follows:

Step 1. Obtain an estimate of the asymptote α_0 by any productive means available.

Step 2. Remove the effects of the asymptote from the data by decreasing each datum by the magnitude of the asymptote.

Step 3. Plot the residuals, obtained in step 2, versus x, on h-cycle base 10 semilogarithmic paper.

Step 4. Fit a straight line by "eye" to as many points as are judged a "good" fit. Start fitting the points in the right-hand end of the plotted curve. Furthermore, after the straight line is fitted, extrapolate back to the semilogarithmic axis to produce an estimate of parameter α_m ; the estimate of the associated β_m is obtained through the use of a simple modified form of the analytic expression for the fitted line.

Step 5. Terminate the "peeling" process if there are fewer than four points to fit; otherwise, proceed to step 6.

Step 6. Obtain a new set of residuals by subtracting the effects of the component fitted in step 4, using original units, from the points that were not included in the fit.

Step 7. Plot the residuals, obtained in step 6, versus x, on semilogarithmic paper. Return to step 4.

Figure 1 depicts an example of component "peeling" by means of the seven aforementioned steps. In this particular instance, four components were extracted from the data after the effects of a constant (asymptote, estimated for this case as two-thirds of the last recorded Y value) had been removed. After the fit of each respective straight line, the parameter estimate of α is obtained by taking $1/100$ (since the original Y values were multiplied by 100 before plotting) the value read off the semilogarithmic axis. The simple formula

$$\hat{\beta} = -(\log\bar{Y} - \log\omega)\ln 10/\bar{x} \qquad (2)$$

was then used to compute the parameter estimate of β when the associated estimate of α became available. The symbols log and ln stand for common and natural logarithms respectively; the single point $(\bar{x},\bar{Y})$ used is any point on the fitted straight line. The estimate of the parameter ω is $\exp(-\hat{\beta})$. Observe the estimates, by component, in the legend of the graph.

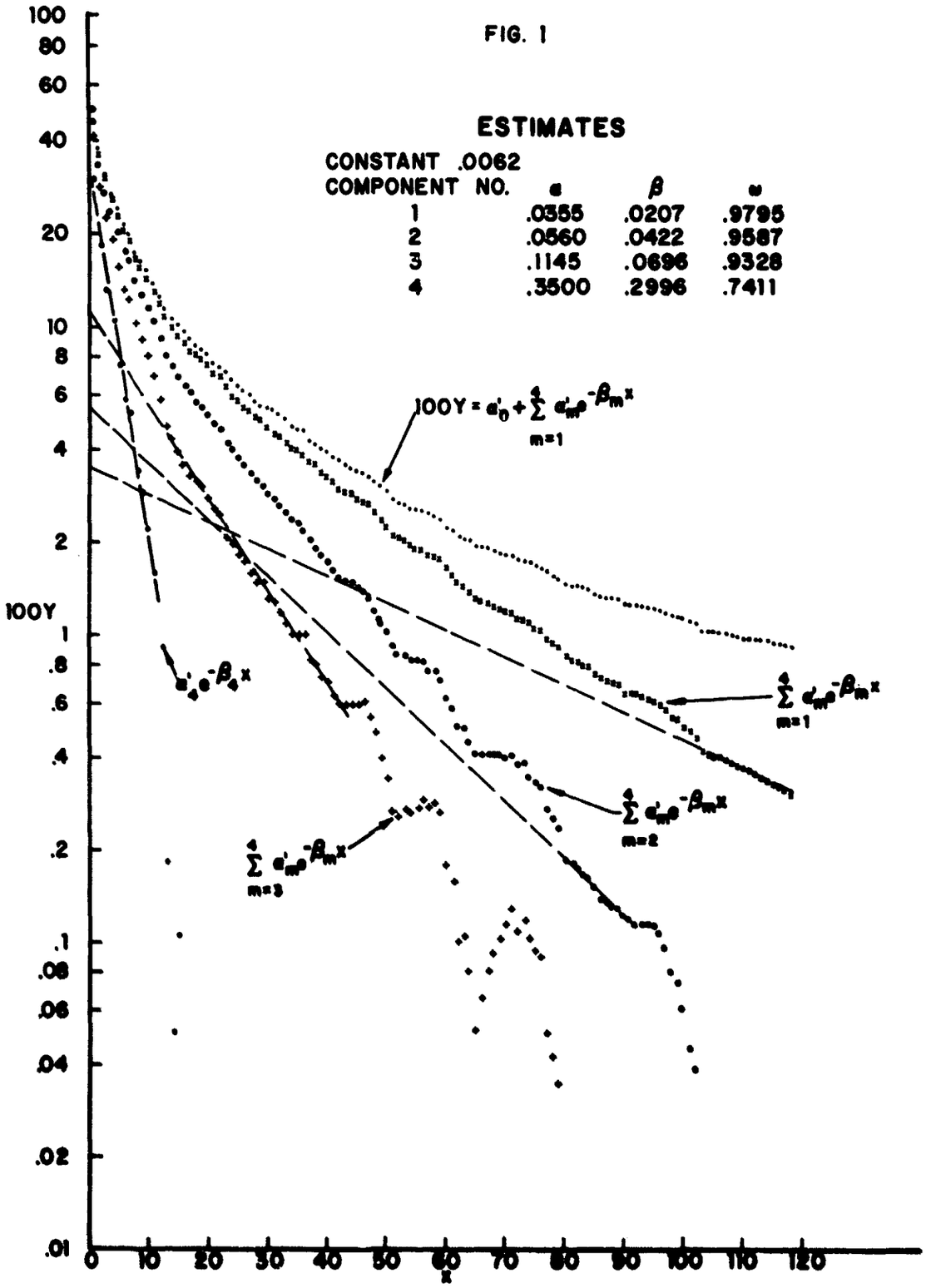
Digital computer version

We now proceed to describe the programmed version of the "peel-off" method. Our description will cover input, general computational steps, fitted straight-line acceptance procedure, additional computations, and output. Following this there will appear a flow chart of the program along with its description, subprogram usage, computer memory requirements, and illustrative numeric examples of program yield in the analysis of simulated and empirical data.

Input:

1. Program parameters (from a punched card).
   a. EXP: Experiment number.
   b. SAMP: Sample number.

FIG. 1


ESTIMATES
CONSTANT .0062
COMPONENT NO. α β ω
1 .0355 .0207 .9795
2 .0560 .0422 .9587
3 .1145 .0696 .9328
4 .3500 .2996 .7411
100Y = a'0 + Σ a'mρ^-βmx (m=1 to 4)
a'4ρ^-β4x
Σ a'mρ^-βmx (m=1 to 4)
Σ a'mρ^-βmx (m=2 to 4)
Σ a'mρ^-βmx (m=3 to 4)
100Y
x
100
80
60
40
20
10
8
6
4
2
1
.8
.6
.4
.2
.1
.08
.06
.04
.02
.01
0
10
20
30
40
50
60
70
80
90
100
110
120

c. L: Sample size.

d. N: Number of components expected to be fitted.

e. $\hat{\alpha}_0$: Estimate of the asymptote.

2. Data (from punched cards).

x,Y values.

Remarks: When the asymptote is absent from the mathematical model, its estimate is zero; otherwise, obtain a non-zero estimate that is based on the experimenter's personal experience, a graphical approach, or any other means that will provide a satisfactory preliminary estimate of this parameter.

General computational steps:

Step 1. Obtain an estimate of the asymptote.

Step 2. Remove the effects of the asymptote from the data.

$$RY_i = Y_i - \hat{\alpha}_0 , \quad i = 1,2,\ldots,L . \tag{3}$$

Step 3. Rearrange the residuals obtained in step 2,

$$(x_{L+1-i}, RY_{L+1-i}) \rightarrow (Z_i, R_i) \tag{4}$$

and transform the residuals R_i logarithmically,

$$G_i = Ln(R_i), \quad i = 1,2,\ldots,L . \tag{5}$$

Step 4. Fit a straight line, by principle of least squares, to K points (Z_i, G_i). Start fitting with $i = 1$. (See line-fitting acceptance procedure following step 7 below.)

Use

$$G = Ln(\alpha) - \beta Z , \quad Z_1 \leq Z \leq Z_K \tag{6}$$

to obtain estimates of α, β for the mth component.

Step 5. Terminate the "peeling" process if there are fewer than four points remaining after deleting the K points used in step 4; i.e., we must have

$$NR = (NPR - K) < 4 \tag{7}$$

where

NPR: Total number of points available for fitting process in step 4 (initially NPR = L).

Otherwise proceed to step 6.

Step 6. Obtain a new set of residuals by removing the effects of the mth component "peeled" in step 4.

$$RP_j = R_j - \hat{\alpha}_m \exp(-\hat{\beta}_m Z_j) , \quad j = K+1, K+2, \ldots, NPR . \tag{8}$$

Step 7 Delete the K elements used in step 4 by repositioning the G,R,Z-arrays

$$Z_j = Z_{K+j} \tag{9}$$

$$R_j = RP_{K+j} \tag{10}$$

$$G_j = \mathrm{Ln}(R_j) , \quad j = 1,2,\ldots,NR . \tag{11}$$

Set NPR = NR and return to step 4.

In the acceptance of the existence of a "real" component which will be discussed next, the motivation for all the steps will not be given, for the whole procedure is subjective. The automation of this scheme is based, primarily, on our experience with one kind of data—nitrogen washout of the lung. It is hoped, however, that the procedure will be fairly general and that the acceptance tests to be described below will have some face validity to the reader. A slight note is occasionally made to motivate a test, but there is no claim for this being the best procedure that can be developed for the "peel-off" method. It has undergone several revisions in our laboratory and must be considered as a best effort at this point in time.

Acceptance procedure:

A fitted least-squares line, which implies the existence of a real component, except for the "last component," is accepted if the following tests are satisfied in the stated order:

1. Runs test.

The differences for the next four successive points are all positive.

$$\begin{aligned} \mathrm{DIFF}_j &= R_j - \hat{\alpha}_m \exp(-\hat{\beta}_m Z_j) \\ &= R_j - \mathrm{YEST}_j > 0, j = K+1, K+2, \ldots, K+4 . \end{aligned} \tag{12}$$

2. Beta test

The slope of the least squares line fitted to the (m-1)st interval is less than the slope of the line fitted to the mth interval.

$$\hat{\beta}_m > \hat{\beta}_{m-1} , \tag{13}$$

where

$$\hat{\beta}_0 = 0.0$$

3. F test.

The ratio is

$$F = \frac{\mathrm{DEVINC}/4}{\mathrm{DEVFIT}/(K-2)} > \mathrm{TOL} , \tag{14}$$

where

$$\text{DEVINC} = \sum_{j=K+1}^{K+4} (R_j - \text{YEST}_j)^2$$

$$\text{DEVFIT} = \sum_{j=1}^{K} (R_j - \text{YEST}_j)^2$$

and

K: Number of points in the interval of fit,

TOL: Upper 1% point of the Snedecor-Fisher (F) distribution with 4 and K-2 degrees of freedom, for $K \leq 32$,

: 4, for $K > 32$.

4. Remaining residuals test.

The differences for all successive residuals not included in the fit are positive.

$$\text{RES}_j = R_j - \text{YEST}_j > 0, \quad j = K+1, K+2, \ldots . \text{NPR} , \tag{15}$$

where

NPR: Number of residuals yet to be fitted.

Note that test 1 is, in general, a subtest of this test. It is used as a quick or preliminary test. This more extensive test 4 is needed to overcome certain abnormal deviations which test 1 will not detect.

5. Alpha test.

The estimate of α_{m+1}, computed in the fitting of the next four successive points, must be less than or equal to $2Y_1$, i.e.,

$$\hat{a}_{m+1} \leq 2Y_1 . \tag{16}$$

Here Y_1 is the first observation of the original data.

This test is not as obvious as the others. It has been empirically determined to guard against including too many points in the interval of fit of the component that is being currently fitted.

The logic of the fitting process is such that from one up to and including a predetermined number of components is fitted. Furthermore, four points is the initial number as well as the least number of points which is included in a fitted component. Also for greater clarity, two distinct categories of fittings need to be considered: fitting the first through the next to the last component and fitting the last component. Here "last component" is defined as a state that exists when either one of the following two conditions holds:

1. The predetermined number N of components expected to be fitted has been reached.

2. NPR − K ≤ 4.

For either condition 1 or 2 holding, all acceptance procedure tests are ignored when we fit the last component.

1. Fitting the first through the next to the last component:

If test 1 is not satisfied: Then either there was or was not a run of four minus signs. In the case of four minus signs, shift the G,R, and Z-arrays such that for each array the (i + 1)st element replaces the ith element: refit a least-squares line to interval of K points of repositioned G, Z-arrays. Repeat this point deletion action (referred to as "creeping") and refitting process until test 1 is satisfied. In the other case, an additional point is included in the interval and the least-squares line is refitted. The process is continued until test 1 is satisfied. This test is admittedly arbitrary and can be altered so that one is more or less certain of detecting the beginning of another component.

If test 2 is not satisfied: Then "creep" in the manner described above for test 1. The "creeping" and line refitting process continues until test 2 is satisfied.

If test 3 is not satisfied: Proceed in the same manner as for test 1 for the case in which a run of four minus signs did not occur.

If test 4 is not satisfied: Repeat action similar to that taken for test 3.

If test 5 is not satisfied: This indicates that too many points were used in the fitting of the least-squares straight line. Proceed to reduce the number of points in the interval of fit one point at a time and use the following criteria for terminating the process:

Test 1, 2, or 4 fails—restore one point to the reduced interval; refit the new interval; accept the fitted line,

or

Test 1, 2, 4 and 5 hold—accept the line fitted to the reduced interval,

or

Number of points in the interval has been reduced to only four—accept the line fitted to these four points.

2. Fitting the last component:

The program will fit the last four points in the sequence of residuals that remains. Again, fitting exactly four points is arbitrary, but it seems to give one essentially unbiased estimates.

Additional computations:

1. Compute a mean square ratio for each component

$$SSR_m = \frac{\sum_{j=1}^{K_m} [(R_j - YEST_j)/YEST_j]^2}{K_m - 2} , \quad m = 1,2,\ldots,M , \tag{17}$$

where

K_m: Number of points used in fitting the mth component,

M : Total number of components fitted.

2. Compute omegas

$$\hat{\omega}_m = \exp(-\hat{\beta}_m) , \quad m = 1,2,\ldots,M . \tag{18}$$

3. Compute estimated Y values

$$YE_i = \hat{a}_0 + \sum_{m=1}^{M} \hat{a}_m \exp(-\hat{\beta}_m x_i) , \quad i = 1,2,\ldots,L . \tag{19}$$

4. Compute ratios

$$RA_i = (Y_i - YE_i)/YE_i , \quad i = 1,2,\ldots,L . \tag{20}$$

5. Compute cumulative sums of squared ratios

$$CRATIO_i = \sum_{j=1}^{i} RA^2_j , \quad i = 1,2,\ldots,L . \tag{21}$$

6. Compute overall "unrefined" mean square ratio

$$PMSR = \frac{CRATIO_L}{L - NP} , \tag{22}$$

where

$$NP = \begin{cases} 2M & \text{for asymptote not estimated,} \\ 2M + 1 & \text{otherwise.} \end{cases}$$

Output:

1. Print—

   a. Experiment number.

   b. Sample number.

   c. For each data point:

      (1) x, Y value.

      (2) Estimated Y value.

      (3) Ratio.

      (4) Cumulative sum of squared ratios.

   d. Model: Model 1 for no asymptote.

      Model 2 otherwise.

   e. Sample size.

   f. Number of components expected to be fitted.

   g. Number of components actually fitted.

   h. Overall "unrefined" mean square ratio.

   i. Estimate of constant.

   j. For each component fitted:

      (1) Range for x values included in the fit.

      (2) Estimate of alpha, beta, and omega.

      (3) Mean square ratio.

2. Write on output tape—

   a. Experiment number.

   b. Sample number.

   c. Model.

   d. Sample size.

   e. Number of parameters.

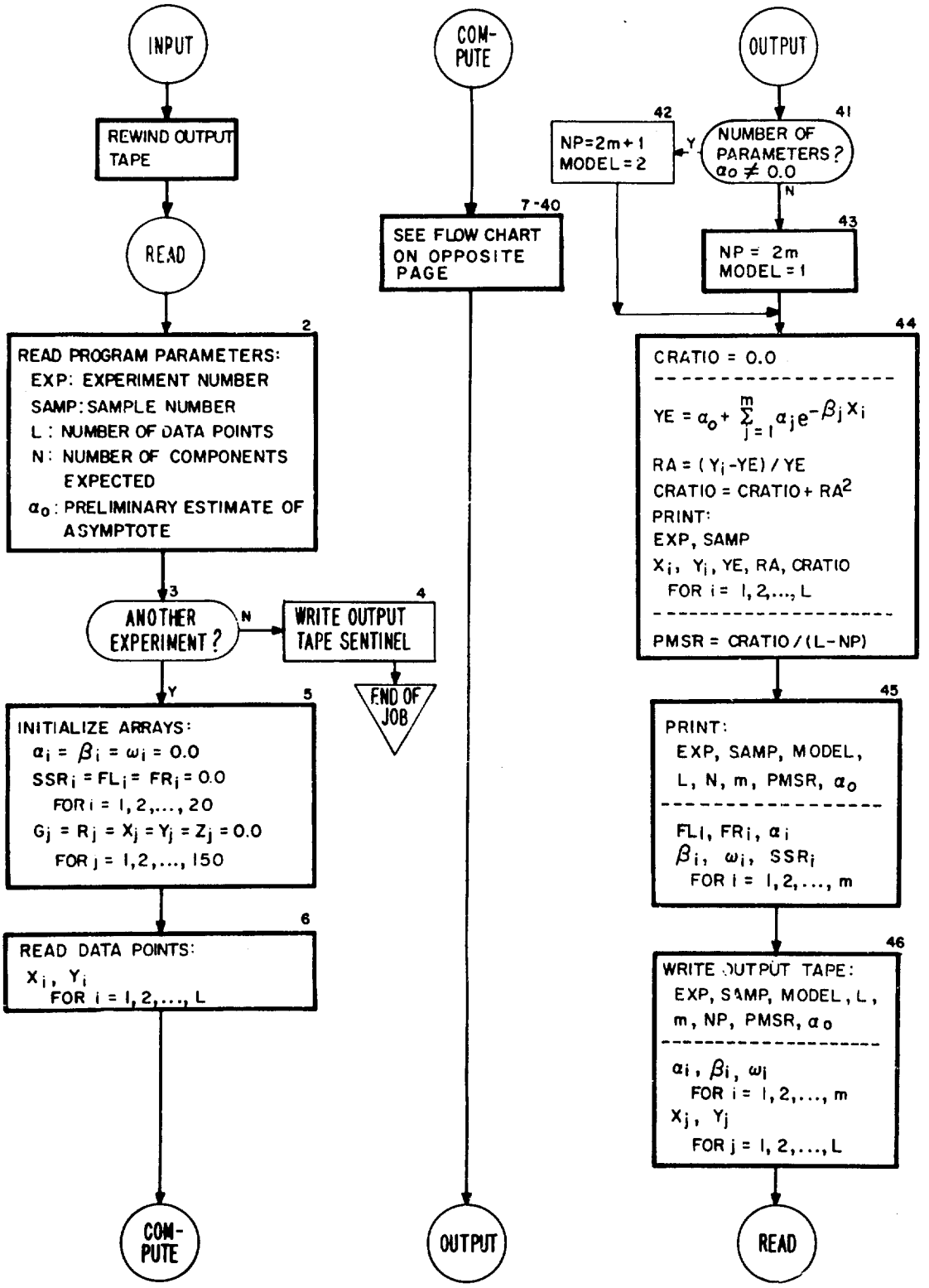
   f. Overall "unrefined" mean square ratio.

   g. The constant and an alpha, beta, and omega for each component fitted.

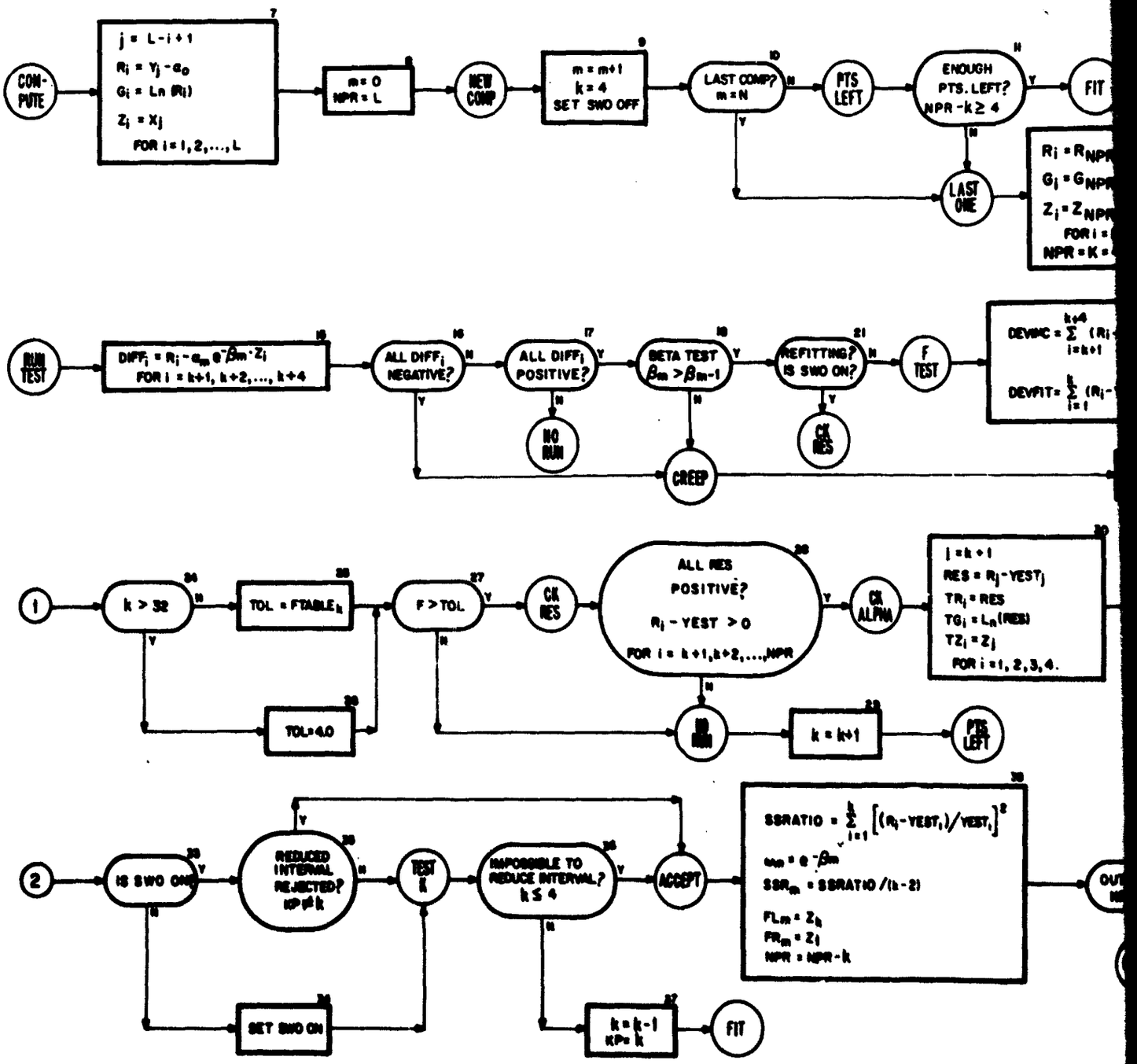
   h. Data points.

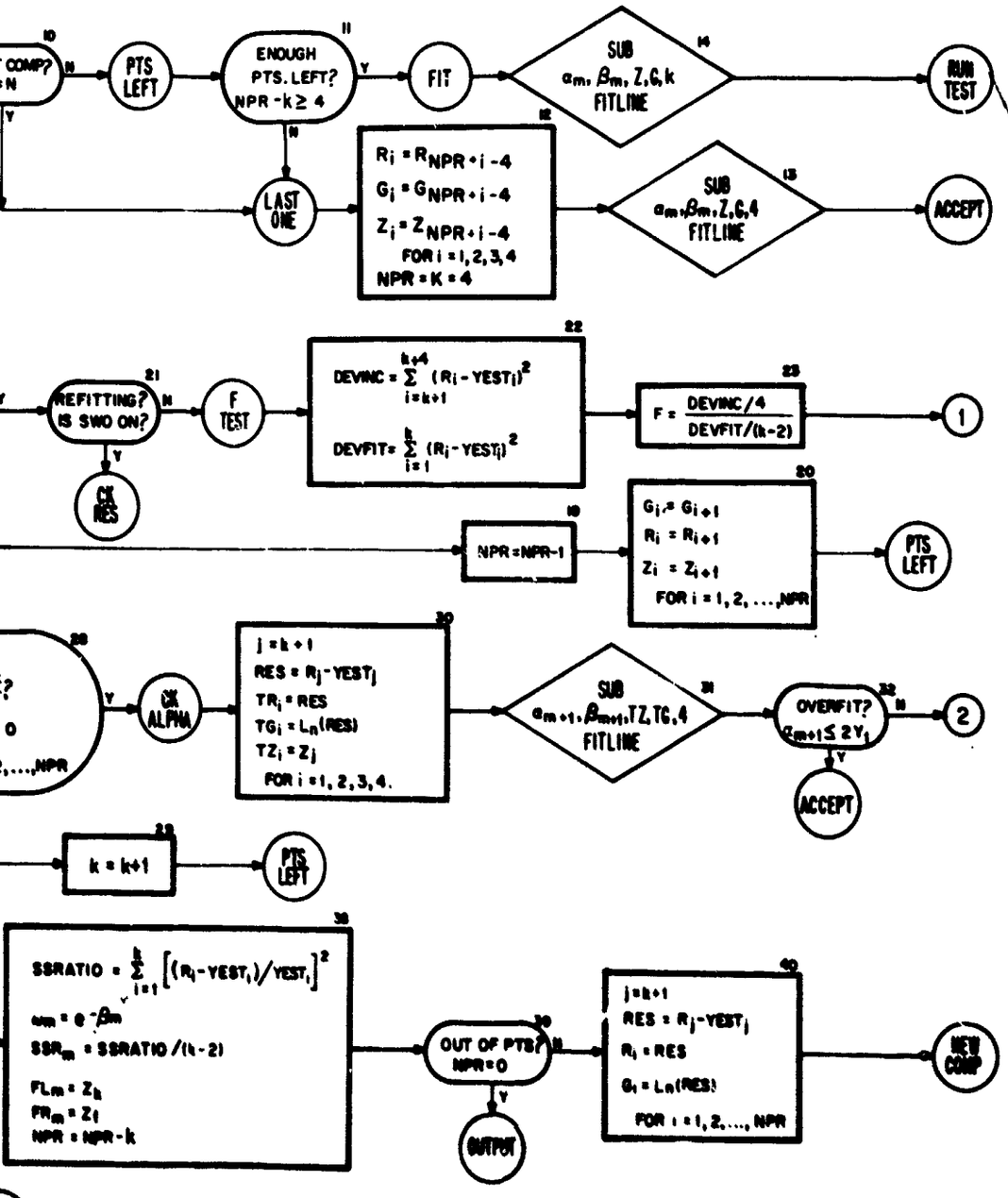
FLOW CHART I

FLOW CHART OF THE "PEEL-OFF" METHOD



FLOW CHART I CONT.




COMP?
N
Y
PTS LEFT
ENOUGH PTS. LEFT? NPR - k ≥ 4
Y
N
FIT
SUB a_m, β_m, Z, G, k FITLINE
RUN TEST
LAST ONE
$R_i = R_{NPR+i-4}$
$G_i = G_{NPR+i-4}$
$Z_i = Z_{NPR+i-4}$
FOR i = 1, 2, 3, 4
NPR = K = 4
SUB $a_m, \beta_m, Z, G, 4$ FITLINE
ACCEPT
REFITTING? IS SWO ON?
F TEST
$DEVINC = \sum_{i=k+1}^{k+4} (R_i - YEST_i)^2$
$DEVFIT = \sum_{i=1}^{k} (R_i - YEST_i)^2$
$F = \frac{DEVINC/4}{DEVFIT/(k-2)}$
1
CK RES
NPR = NPR-1
$G_i = G_{i+1}$
$R_i = R_{i+1}$
$Z_i = Z_{i+1}$
FOR i = 1, 2, ..., NPR
PTS LEFT
CK ALPHA
j = k + 1
RES = R_j - YEST$_j$
TR_i = RES
$TG_i = L_n$(RES)
$TZ_i = Z_j$
FOR i = 1, 2, 3, 4.
SUB $a_{m+1}, \beta_{m+1}, TZ, TG, 4$ FITLINE
OVERFIT? $a_{m+1} \leq 2V_1$
2
ACCEPT
k = k+1
PTS LEFT
$SSRATIO = \sum_{i=1}^{k} [(R_i - YEST_i)/YEST_i]^2$
$a_m = e^{-\beta_m}$
SSR_m = SSRATIO/(k - 2)
$FL_m = Z_k$
$FR_m = Z_1$
NPR = NPR - k
OUT OF PTS? NPR = 0
OUTPUT
j = k+1
RES = R_j - YEST$_j$
R_i = RES
$G_i = L_n$(RES)
FOR i = 1, 2, ..., NPR
NEW COMP

2

Description of flow chart I

Box 1 - 4: Self-explanatory.

5: The following arrays are cleared:

α: Parameter array.

β: Parameter array.

ω: Parameter array.

SSR: Mean squared ratios for the components.

FL: Left-end points of the intervals of component fit.

FR: Right-end points of the intervals of component fit.

X: Abscissae of the data points.

Y: Ordinates of the data points.

G: Natural logarithmic transformation of R-array.

R: Residuals stored in reverse order.

Z: X-array stored in reverse order.

6: Self-explanatory.

7: Residuals obtained by removing the effects of an asymptote, making natural logarithmic transformation of residuals, and developing G, R, and Z-arrays.

8: Initialize component number counter m and number of points remaining to be fitted—gage word NPR.

9: Increment the component number counter m, initialize the number of points to be fitted (K = 4), and set overfit switch (SWO) off.

10: If the component is the last component to be fitted, go to LAST ONE—box 12.

11: If a sufficient number of points remain for a normal fit, go to FIT—box 14.

12: Set up G, R, and Z-arrays so that the last four points can be fitted; set K = 4 and NPR = 4.

13: Fit last component; go to ACCEPT—box 38.

14: A least-squares straight line is fitted to K points.

15: Compute differences for points K + 1 to K + 4.

16: If all differences are negative, go to CREEP—box 19.

17: If all differences are not positive, go to NO RUN—box 29.

18: Beta test. If test holds, go to box 21.

19: Decrement the number of points remaining, NPR.

20: Shift the G, R, Z-arrays such that for each array the (i + 1)st element replaces the ith element; go to PTS LEFT—box 11.

21: If refitting, go to CK RES—box 28.

22-27: F-test. If test fails, go to NO RUN—box 29.

28: If all points not included in the fit lie above the fitted line, go to CK ALPHA—box 30.

29: Increment counter k; go to PTS LEFT—box 11.

30-32: Alpha test. If test holds, go to ACCEPT—box 38.

33: If the overfit switch (SWO) is on, go to box 35.

34: Overfit switch (SWO) was off—set it on; go to TEST K—box 36.

35: If the reduced interval of fit is rejected, accept currently fitted component; go to ACCEPT—box 38.

36: If it is impossible to reduce the current interval of fit by one point, go to ACCEPT—box 38.

37: Current interval of fit can be reduced—reduce counter k by one and set KP equal to new value of counter k; go to FIT—box 14.

38: Compute sum of squared ratios for the K points fitted. Save various quantities associated with the mth fitted component and compute the number of points remaining.

39: If no points remain, go to OUTPUT—box 41.

40: Compute residuals, store in R-array; compute natural logarithms of residuals, store in G-array; reposition Z-array; go to NEW COMP—box 9.

41-43: Compute the number of parameters fitted and set up model number.

44-45: Print output:

1. EXP, SAMP.
2. X, Y, YE, RA, CRATIO, for each data point.
3. EXP, SAMP, MODEL, L, N, M, PMSR.
4. $\hat{a}_0$; FL, FR, SSR, $\hat{a}$, $\hat{\beta}$, $\hat{\alpha}$ for each component fitted.

46: Write output tape:

1. EXP, SAMP, MODEL, L, M, NP, PMSR.
2. $\hat{a}_0$ and estimates $\hat{a}$, $\hat{\beta}$, $\hat{\alpha}$ for each component fitted.
3. X, Y values.

Go to READ—box 2.

Subprogram usage

Function subprograms (the first two are standard library functions) listed below proved helpful:

1. EXPF. Argument: A (location of expression A). Function: computes the value exp (A).

2. LOGF. Argument: A (location of expression A). Function: computes the value ln(A).

3. YEST. Argument: I (location of subscript I). Function: computes the ith estimated Y value using the estimates $\hat{\alpha}_m$ and $\hat{\beta}_m$

A subroutine subprogram needs to be mentioned:

FITLINE. Argument: A (location for storing α), B (location for storing β), C (location of first element in Z-array), D (location of first element in G-array), and J (location of the number of points in the fit). Function: computes estimates of the α and β by fitting a least-squares line to the natural logarithms of J points.

Memory requirements

Program	About 2,680 words
$\hat{\alpha}_0$	1 word
$\hat{\alpha}_1$	M words
$\hat{\beta}_1$	M words
$\hat{\omega}_1$	M words
X-array	L words
Y-array	L words
G-array	L words
R-array	L words
Z-array	L words
Range limits	2M words
Component MSR's	M words
Other	150 words

Total: 2,831 + 5L + 6M words approximately

Examples

Simulated and empirical data served as input to the "peel-off" method computer program. The generation of the simulated data was accomplished through the use of the second equation of 1; the error term $\epsilon(x)$, for each value of x, was produced by means of a subroutine that generated pseudo-random normal deviates with mean = 0, standard deviation = $\rho y(x)$; the generator of the pseudo-random numbers used in the normal deviate generation was of the multiplicative congruential type. The empirical data was restricted to dog lung nitrogen washout data. Program yield for the cases considered is presented in tables I to III for simulated data analysis and tables IV to VI for empirical data analysis.

TABLE I

Preliminary parameter estimates
(Simulated data)

Sample size: 50

Value of ρ: 0.0005

Square root of overall MS ratio: 0.0072

True value of α_0: 0.005 Estimate of α_0: 0.006*

Component No.	MS ratio	Range of x values in interval fitted	α	β	ω
1	12920×10^{-10}	30.0 — 38.0	0.020	0.036	0.965
	True	Value	0.020	0.030	0.970
2	296×10^{-10}	1.0 — 4.0	0.698	0.357	0.700
	True	Value	0.700	0.357	0.700

*Estimate of α_0 arbitrarily chosen in the neighborhood of the true value.

TABLE II

Preliminary parameter estimates
(Simulated data)

Sample size: 75

Value of ρ: 0.0010

Square root of overall MS ratio: 0.0433

True value of α_0: None Estimate of α_0: None

Component No.	MS ratio	Range of x values in interval fitted	α	β	ω
1	856×10^{-10}	71.0 — 75.0	0.065	0.036	0.965
	True	Value	0.040	0.030	0.970
2	31526×10^{-10}	13.0 — 21.0	0.398	0.103	0.902
	True	Value	0.400	0.094	0.910
3	109720×10^{-10}	1.0 — 4.0	0.180	0.481	0.618
	True	Value	0.200	0.431	0.650

TABLE III

Preliminary parameter estimates
(Simulated data)

Sample size: 42
Value of ρ: 0.0030
Square root of overall MS ratio: 0.0125
True value of α_0: 0.050 Estimate of α_0: 0.050*

Component No.	MS ratio	Range of x values in interval fitted	α	β	ω
1	1804×10^{-5}	32.0 — 40.0	0.024	0.084	0.919
	True	Value	0.050	0.105	0.900
2	1968×10^{-5}	18.0 — 28.0	0.042	0.162	0.850
	True	Value	0.100	0.357	0.700
3	39235×10^{-5}	13.0 — 16.0	0.006	0.205	0.814
	True	Value	0.200	0.693	0.500
4	430×10^{-5}	1.0 — 4.0	0.531	0.720	0.487
	True	Value	0.400	1.204	0.300

*Estimate of α_0 simply chosen equal to the true value.

TABLE IV

Preliminary parameter estimates
(Empirical data)

Sample size: 29
Square root of overall MS ratio: 0.0876
Estimate of α_0: 0.005*

Component No.	MS ratio	Range of x values in interval fitted	α	β	ω
1	2018×10^{-9}	26.0 — 29.0	0.012	0.080	0.923
2	25218×10^{-8}	1.0 — 4.0	0.890	0.358	0.699

*Estimate of α_0 based on a visual inspection of plotted curve.

TABLE V

Preliminary parameter estimates
(Empirical data)

Sample size: 49
Square root of overall MS ratio: 0.0665
Estimate of α_0: None

Component No.	MS ratio	Range of x values in interval fitted	α	β	ω
1	186×10^{-10}	46.0 — 49.0	0.015	0.014	0.986
2	4585×10^{-6}	37.0 — 43.0	0.172	0.168	0.850
3	1573×10^{-7}	1.0 — 4.0	0.517	0.362	0.696

TABLE VI

Preliminary parameter estimates

(Empirical data)

Sample size: 140

Square root of overall MS ratio: 0.5504

Estimate of a_0: 0.010*

Component No.	MS ratio	Ranges of x values in interval fitted	α	β	ω
1	268×10^{-10}	105.0 — 117.0	0.005	0.005	0.995
2	800×10^{-6}	77.0 — 88.0	0.013	0.028	0.972
3	2705×10^{-6}	1.0 — 4.0	0.621	0.260	0.771

*Estimate of a_0 based on a visual inspection of plotted curve.

IV. COMPOSITE GAUSS-NEWTON AND GRADIENT METHOD

Marquardt's algorithm

A composite Gauss-Newton and gradient method has been programmed to refine the preliminary estimates of the parameters in our mathematical model that the "peel-off" program provides. It is an iterative method that was designed to eliminate the inadequacies of straight Taylor-series methods and gradient methods but at the same time retain the "good" properties of both approaches. Examples of good properties are: convergence greatly accelerated when close proximity to converged values is attained and region of convergence broader than other methods.

Consider fitting the function

$$Y = f(x;\theta) \tag{23}$$

to a set of data points (x_i, Y_i) $i = 1,2,\ldots L$, $f(x;\theta)$ nonlinear in the parameters represented by the vector $\theta = (\theta_0,\theta_1,\ldots,\theta_{2M})$.

Now, using ratios $\dfrac{Y_i - f(x_i;\theta)}{f(x_i;\theta)}$ to meet our needs, instead of the customary deviations $Y_i - f(x_i;\theta)$, we want to minimize

$$\Phi(\theta) = \sum_{i=1}^{L} [(Y_i - f(x_i;\theta))/f(x_i;\theta)]^2$$

$$= \sum_{i=1}^{L} [Y_i f^{-1}(x_i;\theta) - 1]^2 . \tag{24}$$

Let

$$g(x_i;\theta) = f^{-1}(x_i;\theta) . \tag{25}$$

The Taylor-series expansion of $g(x_i;\theta)$, to first order terms, about the vector $\theta^{(0)} = (\theta_0^{(0)}, \theta_1^{(0)}, \ldots, \theta_{2M}^{(0)})$, vector of preliminary parameter estimates, is

$$g(x_i;\theta) = g(x_i;\theta^{(0)}) + \sum_{j=0}^{2M} D_j g_j(x_i;\theta^{(0)}) , \tag{26}$$

where

$$D_j = \theta_j - \theta_j^{(0)} ,$$

$$g_j(x_i;\theta^{(0)}) = \frac{dg(x_i;\theta)}{d\theta_j} \Big| \theta = \theta^{(0)} .$$

Thus

$$\Phi(\theta) = \sum_{i=1}^{L} \left\{ Y_i[g(x_i;\theta^{(0)}) + \sum_{j=0}^{2M} D_j g_j(x_i;\theta^{(0)})] - 1 \right\}^2 . \tag{27}$$

Partial derivative of $\Phi(\theta)$ with respect to θ_h is

$$\Phi_h(\theta) = 2 \sum_{i=1}^{L} \left\{ Y_i[g(x_i;\theta^{(0)}) + \sum_{j=0}^{2M} D_j g_j(x_i;\theta^{(0)})] - 1 \right\} Y_i g_h(x_i;\theta^{(0)}) . \tag{28}$$

Setting $\Phi_h(\theta) = 0$, simplifying and transposing terms, we have the normal equations

$$\sum_{j=0}^{2M} \left[\sum_{i=1}^{L} Y_i^2 g_h(x_i;\theta^{(0)}) g_j(x_i;\theta^{(0)}) \right] D_j$$

$$= - \sum_{i=1}^{L} [Y_i g(x_i;\theta^{(0)}) - 1] Y_i g_h(x_i;\theta^{(0)}) , \tag{29}$$

$$h = 0,1,\ldots,2M ,$$

or in matrix notation

$$CD = E . \tag{30}$$

Matrix equation 30 is solved for the correction vector, D which in turn is used in obtaining the parameter estimates.

The next trial vector in the iterative process is

$$\theta_j^{(1)} = \theta_j^{(0)} + D_j, \; j = 0,1,\ldots, \tag{31}$$

This is considered the Gauss or Gauss-Newton approach to the estimation problem. In practice, instead of using D_j as is, a step size FD_j, $0 < F \leq 1$, is used in an effort to increase convergence. This is in contrast to the gradient methods that use full steps in the direction of the negative gradient.

Proceeding further to the basic construction of Marquardt's algorithm, equation 30 is modified by adding an arbitrary constant λ to each diagonal element of C. Then we have for the rth iteration, the matrix equation

$$[C^{*(r)} + \lambda^{(r)} I] D^{*(r)} = E^{*(r)} , \tag{32}$$

which is solved for the column matrix $D^{*(r)}$, where

$$C^{*(r)} = [c^{*(r)}_{h\,j}]$$

$$= [c^{(r)}_{h\,j} / (c^{(r)}_{h\,h})^{1/2} (c^{(r)}_{j\,j})^{1/2}] , \tag{33}$$

$$E^{*(r)} = [e^{*(r)}_{j}]$$

$$= [e^{(r)}_{j} / (c^{(r)}_{j\,j})^{1/2}] , \tag{34}$$

and $c_{h\,j}$ and e_j are elements of C and E of equation 30.

From $D^{*(r)}$ we obtain

$$D^{(r)}_{j} = D^{*(r)}_{j} / (c^{(r)}_{j\,j})^{1/2} , \quad j = 0,1,\ldots,2M . \tag{35}$$

The next vector used in the process has components

$$\theta^{(r+1)}_{j} = \theta^{(r)}_{j} + D^{(r)}_{j} , \quad j = 0,1,\ldots,2M \tag{36}$$

The choice of the constant $\lambda^{(r)}$ is critical. By trial and error it is determined so that $\Phi^{(r)} < \Phi^{(r-1)}$.

The strategy employed, along lines somewhat similar to those laid down by Marquardt, is as follows:

Let $\lambda^{(r-1)}$ denote the value of λ that is associated with the (r-1)st iteration, initial values $\lambda^{(0)} = 10^{-2}$ and $\Phi^{(0)} = (PMSR)(DF)$. Choose a constant $\nu > 1$, a suitable value chosen is 10, and

Test 1. If $\lambda^{(r-1)} \leq 10^{-7}$, proceed to test 2. If not, let $\lambda^{(r)} = \lambda^{(r-1)}/\nu$ and compute $\Phi^{(r)}(\lambda^{(r)})$.

If $\Phi^{(r)} < \Phi^{(r-1)}$, then the parameter estimates are accepted; otherwise, proceed to test 2.

Test 2. Let $\lambda^{(r)} = \lambda^{(r-1)}$ and compute $\Phi^{(r)}(\lambda^{(r)})$.

If $\Phi^{(r)} < \Phi^{(r-1)}$, then the parameter estimates are accepted; contrarily, set $\omega = 1$ and proceed to test 3.

Test 3. If $\lambda^{(r)} \leq 10^{10}$, the process is considered to be diverging and further cycling is terminated. When the inequality does not hold, we let $\lambda^{(r)} = \lambda^{(r-1)}\nu^{\omega}$ and compute $\Phi^{(r)}(\lambda^{(r)})$.

If $\Phi^{(r)} < \Phi^{(r-1)}$, then the parameter estimates are accepted; otherwise, proceed to test 4. It should be observed that the values of $\lambda^{(r)}$ can become extremely large in the case of parameter estimates with high (> .99) correlation.

Test 4. Compute an angle

$$\gamma^{(r)} = \cos^{-1}\left[\sum_{j=0}^{2M} d_j^{(r)} e^{(r)} \Big/ \left(\sum_{j=0}^{2M} d_j^{2(r)}\right)^{1/2} \left(\sum_{j=0}^{2M} e_j^{2(r)}\right)^{1/2}\right] . \tag{37}$$

If $\gamma^{(r)} \geqq \gamma_0 = \pi/4$, then increment ω by 1 and return to test 3.

If $\gamma^{(r)} < \gamma_0$, then proceed to find a constant $F^{(r)}$ such that $F^{(r)}D^{(r)}$, $0 < F^{(r)} \leq 1$, is the step vector; i.e., we want

$$\theta^{(r)} = \theta^{(r-1)} + F^{(r)}D^{(r)} . \tag{38}$$

This constant $F^{(r)}$ is found by raising the fraction $\frac{1}{2}$ to successive powers until either $F^{(r)} \leq 10^{-10}$, in which case ω is incremented by 1 and a return to test 3 is made after setting $F^{(r)} = 1$, or it is found that $\Phi^{(r)} < \Phi^{(r-1)}$ and the parameter estimates are accepted.

For use further on in the sequel, we have

$$g(x_i;\theta) = [a_0 + \sum_{m=1}^{M} a_m \exp(-\beta_m x_i)]^{-1} ,$$

$$= S^{-1}(x_i;a,\beta) , \tag{39}$$

$$g_h(x_i;\theta) = -S^{-2}(x_i;a,\beta) S_h(x_i;a,\beta) , \tag{40}$$

$$h = 0,1,\ldots,2M ,$$

where

$$S_h(x_i;a,\beta) = \begin{cases} 1 & , h = 0 \\ \exp(-\beta_{\frac{h+1}{2}} x_i) & , h = 1,3,\ldots,\text{odd integer} \\ -a_{h/2} x_i \exp(-\beta_{h/2} x_i) & , h = 2,4,\ldots,\text{even integer} \end{cases}$$

and α, β are vectors defined as:

$$a = (a_0, a_1, \ldots, a_M) ,$$

$$\beta = (\beta_1, \beta_2, \ldots, \beta_M) .$$

Hence the normal equations 29 become

$$\sum_{j=0}^{2M}\left[\sum_{i=1}^{L} Y_i^2 S^{-4}(x_i;a^{(0)},\beta^{(0)}) S_h(x_i;a^{(0)},\beta^{(0)}) S_j(x_i;a^{(0)},\beta^{(0)})\right] D_j$$

$$= \sum_{i=1}^{L} [Y_i S^{-1}(x_i;a^{(0)},\beta^{(0)}) - 1] Y_i S^{-2}(x_i;a^{(0)},\beta^{(0)}) S_h(x_i;a^{(0)},\beta^{(0)}) , \tag{41}$$

$$h = 0,1,\ldots,2M .$$

Computer version

We continue the sequel with a description of the computer program based on Marquardt's algorithm. This description will include program input, general computational steps, additional computations, and output. Then follows, for completeness, a program flow chart, description of flow chart, subprogram usage, computer memory requirements, and several numeric examples of analyses of both simulated and empirical data.

Input:

1. Program parameters.

   a. From a punched card.

      NIA: Maximum number of iterations allowed.

   b. From output tape produced by "peel-off" program.

      (1) EXP: Experiment number.

      (2) SAMP: Sample number.

      (3) MODEL: 1 for no asymptote,
                 2 otherwise.

      (4) L: Sample size.

      (5) M: Number of components to refine.

      (6) NP: Number of parameters to refine.

      (7) PSMR: "Unrefined" mean square ratio based on preliminary parameter estimates.

2. Preliminary estimates of the model parameters (from output tape).

$$a_0^{(0)}, a_m^{(0)}, \beta_m^{(0)}, \quad m = 1,2,\ldots,M .$$

3. Data (from output tape).

   x, Y values.

General computational steps:

Step 1. Compute (initial) gage quantity

$$\phi^{(0)} = (\text{PMSR})(\text{DF}) , \tag{42}$$

where

$$\text{DF} = L - 2M \text{ for Model 1}$$
$$= L - (2M + 1) \text{ for Model 2.}$$

Step 2. Compute the elements of the matrices $C^{*(r)}$, $E^{*(r)}$,

$$c^{*(r)}_{hj} = c^{(r)}_{hj} \; / \; (c^{(r)}_{hh})^{\frac{1}{2}} \; (c^{(r)}_{jj})^{\frac{1}{2}} \; , \tag{43}$$

where

$$c^{(r)}_{hj} = \sum_{i=1}^{L} Y_i^{*} S^{-4}(x_i;\alpha^{(r)},\beta^{(r)}) S_h(x_i;\alpha^{(r)},\beta^{(r)}) S_j(x_i;\alpha^{(r)},\beta^{(r)}) \; ,$$
$$h,j = 0,1,\ldots,2M \; ,$$

and $S(x_i;\alpha^{(r)},\beta^{(r)})$, $S_h(x_i;\alpha^{(r)},\beta^{(r)})$ are as defined in equations 39, 40.

$$e^{*(r)}_{j} = e^{(r)}_{j} \; / \; (c^{(r)}_{jj})^{\frac{1}{2}} \; , \tag{44}$$

where

$$e^{(r)}_{j} = \sum_{i=1}^{L} [Y_i S^{-1}(x_i;\alpha^{(r)},\beta^{(r)}) - 1] Y_i S^{-2}(x_i;\alpha^{(r)},\beta^{(r)}) \, S_j(x_i;\alpha^{(r)},\beta^{(r)}) \; ,$$
$$j = 0,1,\ldots,2M \; .$$

Step 3. Solve matrix equation

$$[C^{*(r)} + \lambda^{(r)} I] \, D^{*(r)} = E^{*(r)} \; . \tag{45}$$

Step 4. Apply strategy, outlined by series of tests given above, for choosing $\lambda^{(r)}$ and selecting $F^{(r)}$.

Step 5. Terminate process in accordance with tests:

$$|D_j^{(r)}| / (\tau + |\theta_j^{(r)}|) < \epsilon \; , \quad j = 0,\ldots,2M \tag{46}$$

where

$$\theta^{(r)} = (\alpha_0^{(r)} \; , \; \alpha_1^{(r)} \; , \; \beta_1^{(r)} \; , \; \ldots \; , \; \alpha_M^{(r)} \; , \; \beta_M^{(r)}) \; ,$$

τ : arbitrary, say 10^{-3} ,

ϵ : arbitrary, say $5(10^{-7})$,

or

$$N = NIA \tag{47}$$

where

N : Number of completed iterations,

NIA: Maximum number of iterations allowed.

Step 6. Return to step 2, if neither test in step 5 holds, with the current solution vector

$$\theta^{(r)} = (\alpha_0^{(r)} \; , \; \alpha_1^{(r)} \; , \; \beta_1^{(r)} \; , \; \ldots \; , \; \alpha_M^{(r)} \; , \; \beta_M^{(r)}) \tag{48}$$

as the initial vector along with $\lambda^{(r)}$ value for the next iteration.

Additional computations:

1. Compute omegas

$$\omega_m^{(n)} = \exp(-\beta_m^{(n)}), \quad m = 1,2,\ldots,M, \tag{49}$$

where

$\beta_m^{(n)}$ = Final value of β for mth component.

2. Compute estimated Y values

$$YEST_i = \alpha_0^{(n)} + \sum_{m=1}^{M} \alpha_m^{(n)} \exp(-\beta_m^{(n)} x_i), \quad i = 1,2,\ldots,L. \tag{50}$$

3. Compute ratios

$$RA_i = (Y_i - YEST_i) / YEST_i, \quad i = 1,2,\ldots,L. \tag{51}$$

4. Compute cumulative sums of squared ratios

$$CRATIO_i = \sum_{j=1}^{i} RA_j^2, \quad i = 1,2,\ldots,L. \tag{52}$$

5. Compute "refined" mean square ratio

$$FMSR = CRATIO_L/DF, \tag{53}$$

where

DF = L-2M for Model 1
= L − (2M + 1) for Model 2.

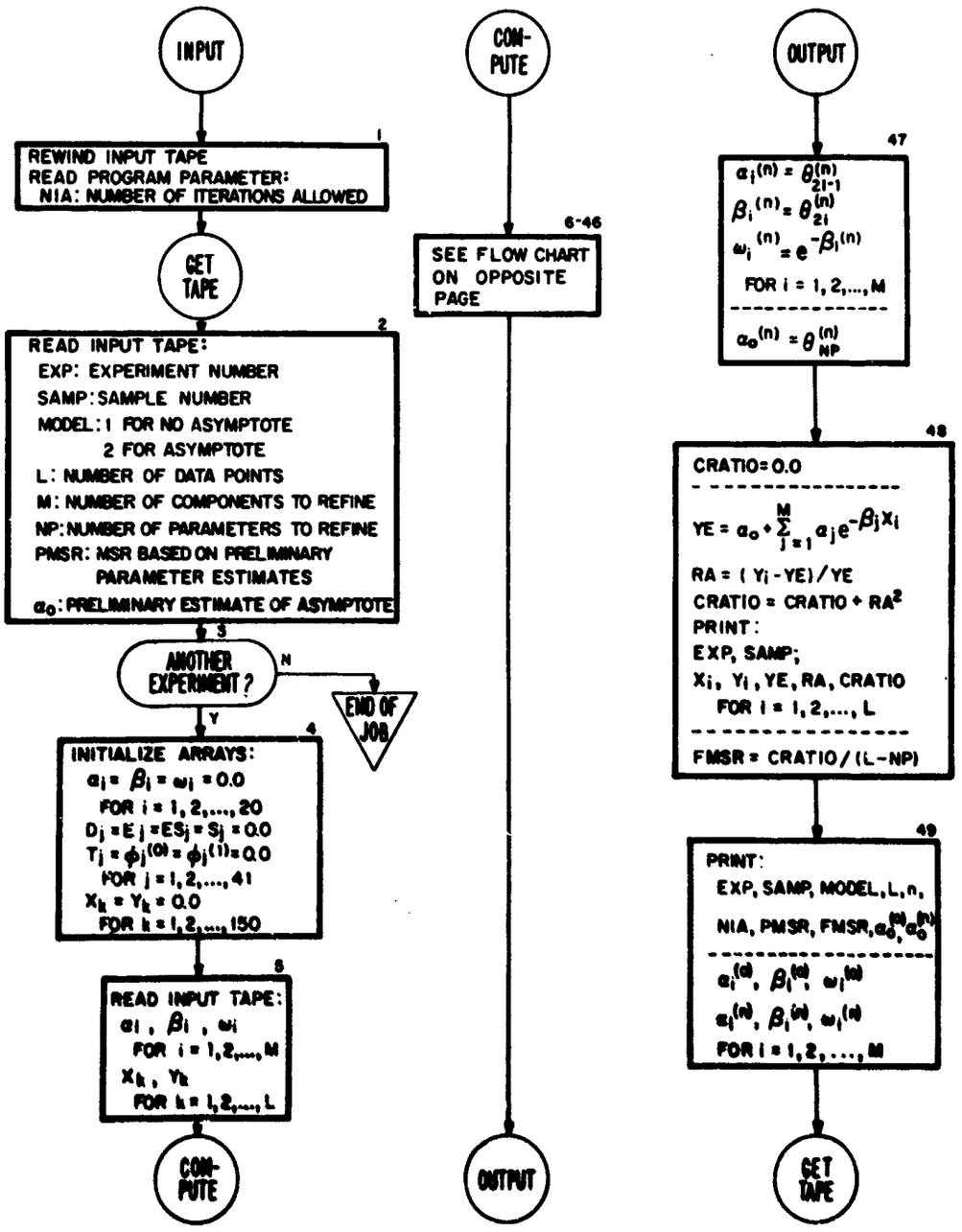
Output:

1. Experiment number.
2. Sample number
3. Data.
4. Estimated Y values.
5. Ratios.
6. Cumulative sums of squared ratios.

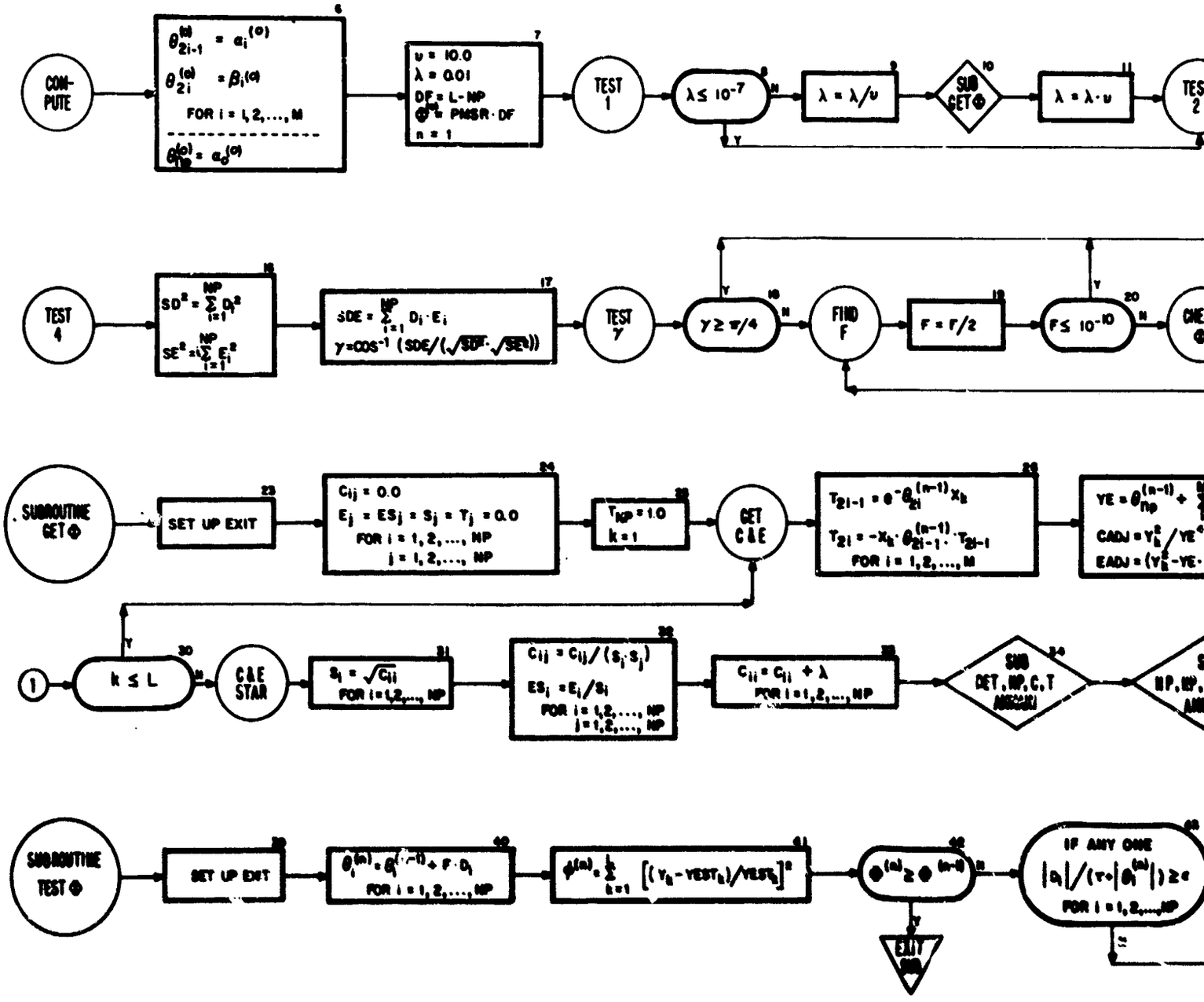
7. Model.
8. Number of data points.
9. Number of iterations taken.
10. Number of iterations allowed.
11. "Unrefined" mean square ratio.
12. "Refined" mean square ratio.
13. Preliminary and final estimates of constant, alphas, betas, and omegas.

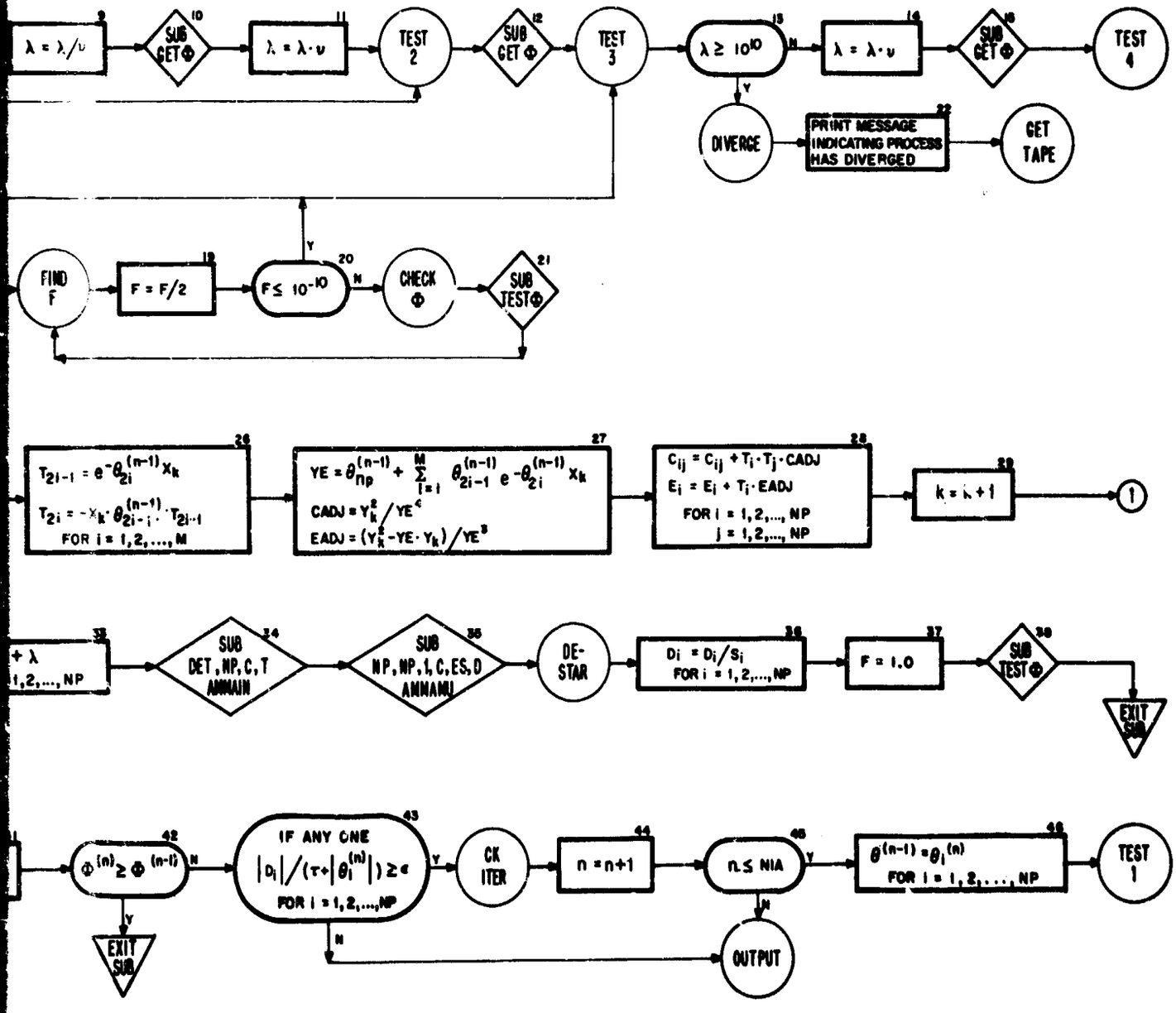
FLOW CHART II

FLOW CHART OF THE COMPOSITE GAUSS–NEWTON AND GRADIENT METHOD



FLOW CHART II CONT.




λ = λ/υ
SUB GET Φ
λ = λ·υ
TEST 2
SUB GET Φ
TEST 3
λ ≥ 10^10
N
λ = λ·υ
SUB GET Φ
TEST 4
Y
DIVERGE
PRINT MESSAGE INDICATING PROCESS HAS DIVERGED
GET TAPE
FIND F
F = F/2
F ≤ 10^-10
Y
N
CHECK Φ
SUB TEST Φ

T2i-1 = e^-θ2i^(n-1) Xk
T2i = -Xk · θ2i-1^(n-1) · T2i-1
FOR i = 1,2,...,M
YE = θnp^(n-1) + Σ θ2i-1^(n-1) e^-θ2i^(n-1) Xk
CADJ = Yk^2 / YE^C
EADJ = (Yk^2 - YE · Yk) / YE^3
Cij = Cij + Ti · Tj · CADJ
Ei = Ei + Ti · EADJ
FOR i = 1,2,..., NP
j = 1,2,..., NP
k = k+1
1

+ λ
1,2,..., NP
SUB DET, NP, C, T AMMAIN
SUB NP, NP, I, C, ES, D AMMAMU
DE-STAR
Di = Di/Si
FOR i = 1,2,..., NP
F = 1.0
SUB TEST Φ
EXIT SUB

Φ(n) ≥ Φ(n-1)
N
Y
EXIT SUB
IF ANY ONE
|Di| / (τ + |θi(n)|) ≥ ε
FOR i = 1,2,...,NP
Y
N
CK ITER
n = n+1
n ≤ NIA
Y
N
OUTPUT
θ(n-1) = θi(n)
FOR i = 1,2,..., NP
TEST 1

2

Description of flow chart II

Box 1 - 3: Self-explanatory.

4: The following arrays are cleared—

α: Parameter array.

β: Parameter array.

ω: Parameter array.

$\theta^{(n-1)}$: (n—1)st parameter iterant array.

$\theta^{(n)}$: nth parameter iterant array.

D: Parameter step adjustments matrix.

E: A basic column matrix used in the iterative process. Its companion matrix is basic matrix C.

S: Square roots of the diagonal elements of matrix C.

ES: Column matrix produced by dividing elements of matrix E by the corresponding elements of S-array.

T: First partial derivatives of the mathematical model with respect to its parameters.

X: Abscissae of the data points.

Y: Ordinates of the data points.

5 - 7: Self-explanatory.

8: Test parameter λ . If the inequality holds, go to TEST 2—box 12.

9: Self-explanatory.

10: See description of subroutine GET Φ, boxes 23-38.

11: Self-explanatory.

12: See description of subroutine GET Φ, boxes 23-38.

13: Test parameter λ . If the inequality holds, go to DIVERGE—box 22.

14: Self-explanatory.

15: See description of subroutine GET Φ, boxes 23-38.

16-18: Compute an angle γ . If the inequality holds, go to TEST 3—box 13.

19: Self-explanatory.

20: Test step adjustment factor F. If the inequality holds, go to TEST 3—box 13.

21: See description of subroutine TEST Φ, boxes 39-46.

22: Self-explanatory.

23: Self-explanatory.

24: Clear locations for elements of matrices C, E, ES, S and T.

25: Compute first derivative of the function in the model with respect to the constant term. Counter k is set equal to one.

26-28: Construction of basic matrices C and E.

29: Self-explanatory.

30: Test counter k against gage word L. If the inequality holds, go to GET C and E—box 26.

31: Compute adjustment quantities for matrices C and E.

32: Transform matrix C into simple correlation matrix, resulting matrix again called C; adjust matrix E, resulting matrix called ES.

33: Add λ to the main diagonal elements of simple correlation matrix C.

34: Compute inverse of modified simple correlation matrix C.

35: Compute product matrix D, $D = C^{-1} ES$.

36: Transform elements of matrix D* back into original units of measurement.

37: Self-explanatory.

38: See description of subroutine TEST Φ, boxes 39-46.

39: Self-explanatory.

40: Compute the nth iterant of parameter estimates.

41: Evaluate nth iterant Φ function.

42: Test (n−1)st and nth iterant Φ functions. If the inequality holds, return to location specified by subroutine exit.

43: Test parameter estimates for acceptability. If the inequality fails to hold, go to OUTPUT—box 47.

44: Self-explanatory.

45: Test number of iterations completed counter n. If the inequality does not hold, go to OUTPUT—box 47.

46: Initialize $\theta^{(n-1)}$-array for next iteration, go to TEST 1—box 8.

47: Relocation of the final estimate of a_0, α's, and β's; computation of the associated ω estimates.

48-49: Print output:

a. EXP. SAMP.

b. X, Y, YE, RA, CRATIO for each data point.

c. EXP, SAMP, MODEL, L, N, NIA, PMSR, FMSR.

d. $a_0^{(0)}$ and estimates $\alpha^{(0)}$, $\beta^{(0)}$, $\omega^{(0)}$ for each component (preliminary estimates).

e. $a^{0(n)}$ and estimates $\alpha^{(n)}$, $\beta^{(n)}$, $\omega^{(n)}$ for each component (final estimates).

Subprogram usage

Function subprograms (the first one is a standard library function) listed below proved very helpful:

1. EXPF. Argument: A (location of expression A). Function: computes the value exp (A).

2. YEST. Argument: K (location of subscript K). Function: computes the kth estimated Y value using the parameter estimates $\alpha_0^{(n)}$, $\alpha_i^{(n)}$, and $\beta_i^{(n)}$, $i = 1,2,\ldots,M$; n denotes nth iteration.

Explicitly presented in the flow chart are two subroutine subprograms:

1. GET Φ. Function: computes full step adjustments for the parameter estimates under refinement.

2. TEST Φ. Function: adjusts the parameter estimates using a full or fractional part of the step adjustments and then computes the sum of squared ratios. Performs a number of tests: $\Phi^{(n)}$ tested against $\Phi^{(n-1)}$, determines if parameter estimates produced during the nth iteration meet the acceptance criterion, or if permissible number of iterations criterion is met in case of failure of parameter estimates to satisfy acceptance criterion.

Two standard library subroutine subprograms not explicitly shown in the flow chart but of paramount importance in the iterative process are:

1. AMMAIN. Arguments: DET (location for value of the determinant of the matrix), N (location for dimension of the matrix), C (location of the first word of the matrix to be inverted), and T (location of the first of 2N words used as temporary storage). Function: computes the inverse and determinant of a matrix C in single-precision floating-point arithmetic. The Gauss process of elimination is used.

2. AMMAMU. Arguments: NR (location for row dimension of matrix C), NC (location for column dimension of matrix C), R (location for column dimension of matrix E), C (location of first element of matrix C), E (location of first element of matrix E), D (location of first element of product matrix D). Function: forms the matrix product D = CE.

Memory requirements

Program	About 3,080 words
$\alpha_0^{(0)}$	1 word
$\alpha_1^{(0)}$	M words
$\beta_1^{(0)}$	M words
$\omega_1^{(0)}$	M words
$\theta_1^{(n-1)}$	2M + 1 words
$\theta_1^{(n)}$	2M + 1 words
$\omega_1^{(n)}$	M words
X-array	L words
Y-array	L words
C-array	$(2M + 1)^2$ words
D-array	2M + 1 words
E-array	2M + 1 words
ES-array	2M + 1 words
Other	170 words

Total: $8{,}256 + 14M + 2L + (2M + 1)^2$ words approximately.

Examples

Parameter-estimate input to the composite Gauss-Newton and gradient method computer program consisted of the preliminary estimates shown in tables I to VI. The refined parameter estimates along with supplementary data are presented in tables VII to XII. Here results for simulated data are reserved for tables VII to IX while for empirical data tables X to XII are utilized.

TABLE VII

Refinement of parameter estimates
(Simulated data)

Sample size: 50

Value of ρ: 0.0005

Number of iterations: 5

Square root of MS ratio for "unrefined" parameter estimates: 0.00725

Square root of MS ratio for "refined" parameter estimates: 0.00047

Parameter	True value	Preliminary estimate	Final estimate
a_0	0.00500	0.00600*	0.00497
a_1	0.02000	0.02047	0.02000
β_1	0.03046	0.03561	0.03034
ω_1	0.97000	0.96502	0.97012
a_2	0.70000	0.69812	0.69981
β_2	0.35667	0.35739	0.35659
ω_2	0.70000	0.69950	0.70006

*Estimate of a_0 arbitrarily chosen in the neighborhood of the true value.

Running times—

Preliminary: 0.1 min.

Final : 1.7 min.

The data points listed below were used to derive the results presented in table VII; thus, they may serve as test data.

Test Data

x	1	2	3	4	5
Y	0.514113904	0.366671076	0.263295656	0.190918417	0.139905307
x	6	7	8	9	10
Y	0.104017778	0.078778202	0.061029563	0.048484372	0.039510206
x	11	12	13	14	15

Y	0.033133051	0.028566658	0.025232096	0.022818472	0.020982678
x	16	17	18	19	20
Y	0.019616327	0.018553200	0.017690570	0.016995343	0.016445823
x	21	22	23	24	25
Y	0.015943802	0.015504852	0.015120464	0.014761646	0.014443451
x	26	27	28	29	30
Y	0.014126197	0.013832992	0.013563697	0.013294183	0.013034256
x	31	32	33	34	35
Y	0.012786053	0.012550015	0.012322536	0.012106027	0.011883805
x	36	37	38	39	40
Y	0.011691969	0.011487734	0.011281136	0.011103610	0.010918203
x	41	42	43	44	45
Y	0.010739656	0.010559082	0.010402436	0.010236416	0.010073497
x	46	47	48	49	50
Y	0.009914986	0.009782278	0.009636448	0.009496308	0.009359059

TABLE VIII

Refinement of parameter estimates
(Simulated data)

Sample size: 75
Value of ρ: 0.0010
Number of iterations: 6
Square root of MS ratio for "unrefined" parameter estimates: 0.0433
Square root of MS ratio for "refined" parameter estimates: 0.0008

Parameter	True value	Preliminary estimate	Final estimate
α_0	None	None	None
α_1	0.0400	0.0649	0.0400
β_1	0.0305	0.0358	0.0305
ω_1	0.9700	0.9648	0.9700
α_2	0.4000	0.3977	0.3997
β_2	0.0943	0.1029	0.0943
ω_2	0.9100	0.9022	0.9100
α_3	0.2000	0.1796	0.1967
β_3	0.4308	0.4812	0.4270
ω_3	0.6500	0.6180	0.6525

Running times—
Preliminary: 0.2 min.
Final : 2.4 min.

TABLE IX

Refinement of parameter estimates
(Simulated data)

Sample size: 42
Value of ρ: 0.0030
Number of iterations: 89
Square root of MS ratio for "unrefined" parameter estimates: 0.0125
Square root of MS ratio for "refined" parameter estimates: 0.0028

Parameter	True value	Preliminary estimate	Final estimate
α_0	0.0500	0.0500*	0.0500
α_1	0.0500	0.0243	0.0474
β_1	0.1054	0.0845	0.1028
ω_1	0.9000	0.9190	0.9023
α_2	0.1000	0.0419	0.0549
β_2	0.3567	0.1622	0.2976
ω_2	0.7000	0.8503	0.7426
α_3	0.2000	0.0061	0.2121
β_3	0.6932	0.2055	0.5723
ω_3	0.5000	0.8143	0.5642
α_4	0.4000	0.5314	0.4440
β_4	1.2040	0.7203	1.2233
ω_4	0.3000	0.4866	0.2943

*Estimate of α_0 simply chosen equal to the true value.
Running times—
Preliminary: 1.2 min.
Final : 10.5 min.

TABLE X

Refinement of parameter estimates
(Empirical data)

Sample size: 29
Number of iterations: 82
Square root of MS ratio for "unrefined" parameter estimates: 0.0876
Square root of MS ratio for "refined" parameter estimates: 0.0278

Parameter	Preliminary estimate	Final estimate
α_0	0.0050*	0.0062
α_1	0.0119	0.5566
β_1	0.0797	0.2838
ω_1	0.9234	0.7530
α_2	0.8899	0.3864
β_2	0.3582	0.5698
ω_2	0.6989	0.5657

*Estimate of α_0 based on a visual inspection of plotted curve.
Running times—
Preliminary: 0.1 min.
Final : 3.4 min.

TABLE XI

Refinement of parameter estimates (Empirical data)

Sample size: 49
Number of iterations: 11
Square root of MS ratio for "unrefined" parameter estimates: 0.0665
Square root of MS ratio for "refined" parameter estimates: 0.0218

Parameter	Preliminary estimate	Final estimate
α_0	None	None
α_1	0.0147	0.0107
β_1	0.0139	0.0085
ω_1	0.9862	0.9916
α_2	0.1717	0.1037
β_2	0.1625	0.1143
ω_2	0.8500	0.8920
α_3	0.5173	0.6490
β_3	0.3623	0.3922
ω_3	0.6961	0.6756

Running times—
Preliminary: 0.1 min.
Final : 1.5 min.

TABLE XII

Refinement of parameter estimates (Empirical data)

Sample size: 140
Number of iterations: 15
Square root of MS ratio for "unrefined" parameter estimates: 0.5504
Squsre root of MS ratio for "refined" parameter estimates: 0.0216

Parameter	Preliminary estimate	Final estimate
α_0	0.0100*	0.0116
α_1	0.0047	0.0908
β_1	0.0049	0.0410
ω_1	0.9951	0.9599
α_2	0.0132	0.1695
β_2	0.0279	0.1477
ω_2	0.9725	0.8627
α_3	0.6208	0.4374
β_3	0.2600	0.4729
ω_3	0.7710	0.6232

*Estimate of α_0 based on a visual inspection of plotted curve.
Running times—
Preliminary: 0.3 min.
Final : 4.8 min.

V. COMMENTS

Even a cursory examination of the basic steps pertinent to the "peel-off" method reveals that these steps are quite simple in mathematical content and do not offer any particular problem in programming. The overall "sticky" areas of "when or where to continue or discontinue" the fitting process no doubt can be improved upon. What and how much improvement is a question that can best be answered through extensive testing of the program on data frequently handled at a particular computing installation or laboratory. Needless to say, one should obtain the best possible preliminary estimate of the constant term when such is present in the model being fitted. A poor preliminary estimate of the constant term results in inferior component estimates, and this in turn can result in a great increase in the number of iterations for refinement or the divergence of the refining process. A number of methods for constant-term estimation by computer were tried but results were in general disappointing. Further work along these lines is contemplated. Also some Monte Carlo studies of the parameters estimates are planned. All raw data that is not monotone decreasing should be smoothed before being analyzed and preliminary parameters estimates should always be refined. Lastly the use of the upper 1% points of the Snedecor-Fisher (F) distribution in the F test as proposed and used by the authors may provide a test that is too stringent for some users. In such cases the appropriate percentage points can be selected on the basis of program yield on data commonly handled.

One should be able to program the composite Gauss-Newton and gradient iterative method for nonlinear parameter estimation with relative ease. The method is straightforward, requiring no "special" coding technics. Minor changes envisaged in a few instances are those relative to the parameter values of ϵ, γ_0, $\lambda^{(0)}$, ν, and τ. These changes may be made in accordance with the immediate requirements of the user. In passing, one might consider the use of double-precision floating point in computing Φ with Y_i in single or double precision for tests 1 and 2 (see strategy for choosing λ) if round-off contributes to erratic fluctuations in Φ.

REFERENCES

1. Danford, M. B. Some problems on the use of negative exponential curves in biology. SAM-TR-65-4, Mar. 1965.

2. Marquardt, D. W. An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Indus. Appl. Math. 11 (2):431-441 (June 1963).

3. Philco (a Subsidiary of Ford Motor Company) Computer Division, Philco 2000 (Algebriac Programming Language) ALTAC III, Jan. 1963.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
USAF School of Aerospace Medicine
Aerospace Medical Division (AFSC)
Brooks Air Force Base, Texas

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b GROUP

3. REPORT TITLE
FITTING MULTI-COMPONENT EXPONENTIAL DECAY CURVES BY DIGITAL COMPUTER

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(Last name, first name, initial)*
Bell, Earl L.
Garcia, Raul

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
Aug. 1965	38	3

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	SAM-TR-65-59
c. Task No. 631902	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. AVAILABILITY/LIMITATION NOTICES
Qualified requesters may obtain copies of this report from DDC.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	USAF School of Aerospace Medicine Aerospace Medical Division (AFSC) Brooks Air Force Base, Texas

13. ABSTRACT

The mechanical-graphical "peel-off" method and Marquardt's composite Gauss-Newton and gradient iterative method were programmed for the Philco 2000, a 16K asynchronous digital computer. Both programs were coded in the Philco Algebraic Programming Language (ALTAC) using single-precision floating-point arithmetic.

Background material, flow charts, flow chart descriptions, subprogram usage, computer memory requirements, and illustrative numeric examples of the analyses of both simulated and empirical data are given. Each sample of simulated data possessed an error component; the effects of an asymptote, in several instances, were included during the generation of the data. Dog lung nitrogen washout activity experiments were the source of the empirical data.

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Mathematical sciences Curve fitting Decay curves Exponential decay curves, multi-component Philco ALTAC digital computer programs "Peel-off" method Gauss-Newton and gradient method Digital computer programming Marquardt's algorithm						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.